Math 5320 Project 1

Group: Ryan, Mitchell, Enoch, Michael

2/11/2022

1. Genre: Animation

(Top 10 from AFI and two randomly chosen ones)

```
AFI <- c("SnowWhite", "Pinocchio", "Bambi", "LionKing", "Fantasia", "ToyStory", "Beauty&Beast", "Shrek"
data.frame(AFI)
```

```
##             AFI
## 1     SnowWhite
## 2     Pinocchio
## 3         Bambi
## 4      LionKing
## 5      Fantasia
## 6      ToyStory
## 7  Beauty&Beast
## 8         Shrek
## 9    Cinderella
## 10         Nemo
## 11        WallE
## 12           Up
```

2. Movie Rank by group

```
##       GroupList
## 1     SnowWhite
## 2    Cinderella
## 3  Beauty&Beast
## 4      LionKing
## 5         Shrek
## 6      ToyStory
## 7          Nemo
## 8     Pinocchio
## 9            Up
## 10         Nemo
## 11     Fantasia
## 12        WallE
```

3. Wikipedia reference

Adjacent matrix

```
##              SnowWhite Pinocchio Bambi LionKing Fantasia ToyStory Beauty&Beast
## SnowWhite            0         1     1        1        1        1            1
## Pinocchio            1         0     1        1        1        1            1
## Bambi                1         1     0        1        1        1            1
## LionKing             1         1     1        0        1        1            1
## Fantasia             1         1     1        1        0        1            1
## ToyStory             1         1     1        1        1        0            1
## Beauty&Beast         1         1     1        1        1        1            0
## Shrek                0         0     0        1        0        1            1
## Cinderella           1         1     1        1        1        1            1
## Nemo                 1         1     1        1        1        1            1
## WallE                1         1     1        1        1        1            1
## Up                   1         0     1        1        1        1            1
##              Shrek Cinderella Nemo WallE Up
## SnowWhite        1          1    1     1  1
## Pinocchio        1          1    1     1  1
## Bambi            0          1    1     1  1
## LionKing         1          1    1     1  1
## Fantasia         1          1    1     1  1
## ToyStory         1          1    1     1  1
## Beauty&Beast     1          1    1     1  1
## Shrek            0          0    1     1  0
## Cinderella       1          0    1     1  1
## Nemo             1          1    0     1  1
## WallE            1          1    1     0  1
## Up               1          1    1     1  0
```

Stochastic matrix

```
##              SnowWhite Pinocchio Bambi LionKing Fantasia ToyStory Beauty&Beast
## SnowWhite          0.0   0.11111   0.1 0.090909      0.1 0.090909     0.090909
## Pinocchio          0.1   0.00000   0.1 0.090909      0.1 0.090909     0.090909
## Bambi              0.1   0.11111   0.0 0.090909      0.1 0.090909     0.090909
## LionKing           0.1   0.11111   0.1 0.000000      0.1 0.090909     0.090909
## Fantasia           0.1   0.11111   0.1 0.090909      0.0 0.090909     0.090909
## ToyStory           0.1   0.11111   0.1 0.090909      0.1 0.000000     0.090909
## Beauty&Beast       0.1   0.11111   0.1 0.090909      0.1 0.090909     0.000000
## Shrek              0.0   0.00000   0.0 0.090909      0.0 0.090909     0.090909
## Cinderella         0.1   0.11111   0.1 0.090909      0.1 0.090909     0.090909
## Nemo               0.1   0.11111   0.1 0.090909      0.1 0.090909     0.090909
## WallE              0.1   0.11111   0.1 0.090909      0.1 0.090909     0.090909
## Up                 0.1   0.00000   0.1 0.090909      0.1 0.090909     0.090909
##              Shrek Cinderella     Nemo    WallE  Up
## SnowWhite      0.1        0.1 0.090909 0.090909 0.1
## Pinocchio      0.1        0.1 0.090909 0.090909 0.1
## Bambi          0.0        0.1 0.090909 0.090909 0.1
## LionKing       0.1        0.1 0.090909 0.090909 0.1
## Fantasia       0.1        0.1 0.090909 0.090909 0.1
## ToyStory       0.1        0.1 0.090909 0.090909 0.1
## Beauty&Beast   0.1        0.1 0.090909 0.090909 0.1
## Shrek          0.0        0.0 0.090909 0.090909 0.0
```

```
## Cinderella     0.1        0.0 0.090909 0.090909 0.1
## Nemo           0.1        0.1 0.000000 0.090909 0.1
## WallE          0.1        0.1 0.090909 0.000000 0.1
## Up             0.1        0.1 0.090909 0.090909 0.0
```

Checking the stochastic matrix (All columns sum up to 1)

```
lapply(list(W), colSums, na.rm=TRUE)
```

```
## [[1]]
##    SnowWhite    Pinocchio        Bambi     LionKing     Fantasia     ToyStory
##            1            1            1            1            1            1
## Beauty&Beast        Shrek   Cinderella         Nemo        WallE           Up
##            1            1            1            1            1            1
```

Questions:

(a) Were there dangling nodes?

No. There weren't any columns with zero elements.

(b) Is the network connected?

Yes. The adjacent matrix and the transposed matrix were summed, and used to check the connectivity.

Adjacent + Transposed matrix

```
##              SnowWhite Pinocchio Bambi LionKing Fantasia ToyStory Beauty&Beast
## SnowWhite            0         2     2        2        2        2            2
## Pinocchio            2         0     2        2        2        2            2
## Bambi                2         2     0        2        2        2            2
## LionKing             2         2     2        0        2        2            2
## Fantasia             2         2     2        2        0        2            2
## ToyStory             2         2     2        2        2        0            2
## Beauty&Beast         2         2     2        2        2        2            0
## Shrek                1         1     0        2        1        2            2
## Cinderella           2         2     2        2        2        2            2
## Nemo                 2         2     2        2        2        2            2
## WallE                2         2     2        2        2        2            2
## Up                   2         1     2        2        2        2            2
##              Shrek Cinderella Nemo WallE Up
## SnowWhite        1          2    2     2  2
## Pinocchio        1          2    2     2  1
## Bambi            0          2    2     2  2
## LionKing         2          2    2     2  2
## Fantasia         1          2    2     2  2
## ToyStory         2          2    2     2  2
## Beauty&Beast     2          2    2     2  2
## Shrek            0          1    2     2  1
## Cinderella       1          0    2     2  2
## Nemo             2          2    0     2  2
## WallE            2          2    2     0  2
## Up               1          2    2     2  0
```

(c) Is it regular?

Yes. After powering matrix W by 3000, we could verify that all the columns had similar values with similar patterns.

```
W %^% 3000
```

```
##              SnowWhite Pinocchio    Bambi LionKing Fantasia ToyStory
## SnowWhite     0.088119  0.088119 0.088119 0.088119 0.088119 0.088119
## Pinocchio     0.087237  0.087237 0.087237 0.087237 0.087237 0.087237
## Bambi         0.084447  0.084447 0.084447 0.084447 0.084447 0.084447
## LionKing      0.088853  0.088853 0.088853 0.088853 0.088853 0.088853
## Fantasia      0.088119  0.088119 0.088119 0.088119 0.088119 0.088119
## ToyStory      0.088853  0.088853 0.088853 0.088853 0.088853 0.088853
## Beauty&Beast  0.088853  0.088853 0.088853 0.088853 0.088853 0.088853
## Shrek         0.040388  0.040388 0.040388 0.040388 0.040388 0.040388
## Cinderella    0.088119  0.088119 0.088119 0.088119 0.088119 0.088119
## Nemo          0.088853  0.088853 0.088853 0.088853 0.088853 0.088853
## WallE         0.088853  0.088853 0.088853 0.088853 0.088853 0.088853
## Up            0.079307  0.079307 0.079307 0.079307 0.079307 0.079307
##              Beauty&Beast    Shrek Cinderella     Nemo    WallE       Up
## SnowWhite        0.088119 0.088119   0.088119 0.088119 0.088119 0.088119
## Pinocchio        0.087237 0.087237   0.087237 0.087237 0.087237 0.087237
## Bambi            0.084447 0.084447   0.084447 0.084447 0.084447 0.084447
## LionKing         0.088853 0.088853   0.088853 0.088853 0.088853 0.088853
## Fantasia         0.088119 0.088119   0.088119 0.088119 0.088119 0.088119
## ToyStory         0.088853 0.088853   0.088853 0.088853 0.088853 0.088853
## Beauty&Beast     0.088853 0.088853   0.088853 0.088853 0.088853 0.088853
## Shrek            0.040388 0.040388   0.040388 0.040388 0.040388 0.040388
## Cinderella       0.088119 0.088119   0.088119 0.088119 0.088119 0.088119
## Nemo             0.088853 0.088853   0.088853 0.088853 0.088853 0.088853
## WallE            0.088853 0.088853   0.088853 0.088853 0.088853 0.088853
## Up               0.079307 0.079307   0.079307 0.079307 0.079307 0.079307
```

(d) Is there a unique steady state vector?

The eigenvectors corresponding to the eigenvalue 1 is given as below.

```
eigen(W)$values
```

Eigenvalues

```
##  [1]  1.0000e+00+0.000e+00i -1.1818e-01+1.321e-02i -1.1818e-01-1.321e-02i
##  [4] -1.0000e-01+0.000e+00i -1.0000e-01+0.000e+00i -1.0000e-01+0.000e+00i
##  [7] -1.0000e-01+0.000e+00i -9.0909e-02+0.000e+00i -9.0909e-02+0.000e+00i
## [10] -9.0909e-02+0.000e+00i -9.0909e-02+0.000e+00i -1.4961e-17+0.000e+00i
```

```
W_eigenvectors <- eigen(W)$vectors
W_eigenvectors
```

Eigenvectors

```
##              [,1]                [,2]                [,3]           [,4]
##  [1,] -0.30148+0i  0.055632-0.038684i  0.055632+0.038684i  7.4700e-01+0i
##  [2,] -0.29847+0i  0.100396-0.015850i  0.100396+0.015850i -4.7851e-17+0i
```

```
##  [3,] -0.28892+0i -0.798197+0.000000i -0.798197+0.000000i -6.5362e-01+0i
##  [4,] -0.30399+0i  0.035553-0.035514i  0.035553+0.035514i -8.5175e-16+0i
##  [5,] -0.30148+0i  0.055632-0.038684i  0.055632+0.038684i  1.3339e-02+0i
##  [6,] -0.30399+0i  0.035553-0.035514i  0.035553+0.035514i  3.1582e-16+0i
##  [7,] -0.30399+0i  0.035553-0.035514i  0.035553+0.035514i  9.5702e-17+0i
##  [8,] -0.13818+0i -0.150132+0.119814i -0.150132-0.119814i  2.8830e-16+0i
##  [9,] -0.30148+0i  0.055632-0.038684i  0.055632+0.038684i  1.3339e-02+0i
## [10,] -0.30399+0i  0.035553-0.035514i  0.035553+0.035514i  9.5702e-17+0i
## [11,] -0.30399+0i  0.035553-0.035514i  0.035553+0.035514i -8.6132e-17+0i
## [12,] -0.27133+0i  0.503274+0.189661i  0.503274-0.189661i -1.2005e-01+0i
##                 [,5]           [,6]           [,7]           [,8]
##  [1,]  5.2052e-01+0i -3.0293e-01+0i  2.0409e-01+0i  1.5470e-16+0i
##  [2,] -1.0036e-16+0i -6.4775e-16+0i  2.0520e-15+0i  6.5585e-16+0i
##  [3,] -8.2132e-01+0i -7.0158e-01+0i  4.7735e-01+0i  6.2282e-15+0i
##  [4,] -1.4327e-15+0i -2.1924e-15+0i  2.3383e-15+0i  1.1478e-02+0i
##  [5,] -3.4920e-03+0i  3.0343e-01+0i  8.4240e-02+0i  4.1868e-16+0i
##  [6,]  9.4089e-16+0i  1.5612e-15+0i  8.3510e-17+0i -4.0998e-02+0i
##  [7,] -4.0145e-16+0i  2.1592e-16+0i -4.4141e-16+0i -4.8557e-01+0i
##  [8,]  3.0771e-16+0i  4.6090e-16+0i -7.3668e-16+0i -5.3467e-16+0i
##  [9,]  2.1613e-01+0i  1.5281e-01+0i  8.0978e-02+0i  1.5538e-15+0i
## [10,]  3.8765e-16+0i -5.8962e-16+0i -2.7439e-16+0i  8.1868e-01+0i
## [11,] -9.4089e-17+0i  1.9100e-16+0i -3.9966e-16+0i -3.0359e-01+0i
## [12,]  8.8156e-02+0i  5.4827e-01+0i -8.4666e-01+0i -8.9983e-15+0i
##                 [,9]          [,10]          [,11]          [,12]
##  [1,] -2.8898e-16+0i  7.9009e-16+0i  5.0454e-16+0i -6.2094e-17+0i
##  [2,]  2.9506e-16+0i -1.3073e-15+0i  2.4308e-17+0i -2.0698e-16+0i
##  [3,]  4.0855e-15+0i -1.9460e-14+0i -2.0251e-15+0i -7.0711e-01+0i
##  [4,]  1.7702e-01+0i -6.4873e-02+0i -8.7368e-01+0i  1.1039e-16+0i
##  [5,] -6.4509e-16+0i  2.6213e-15+0i  6.0825e-16+0i  3.4497e-17+0i
##  [6,]  4.4477e-01+0i -6.9366e-01+0i  3.6984e-01+0i -4.8296e-17+0i
##  [7,]  2.6574e-01+0i  6.5625e-01+0i  2.1842e-01+0i -4.8296e-17+0i
##  [8,] -4.2003e-16+0i  1.7880e-15+0i  1.0195e-16+0i  7.0711e-01+0i
##  [9,]  2.3609e-16+0i  1.4758e-16+0i  3.5151e-16+0i  3.4497e-17+0i
## [10,] -5.2379e-02+0i  2.4956e-01+0i  2.1842e-01+0i -4.8296e-17+0i
## [11,] -8.3515e-01+0i -1.4727e-01+0i  6.6996e-02+0i -3.4497e-17+0i
## [12,] -3.6423e-15+0i  1.6753e-14+0i  6.4955e-16+0i  1.4144e-16+0i
```

```
W_ssv <- W_eigenvectors/sum(W_eigenvectors)
W_ssv
```

```
##             [,1]                [,2]                [,3]           [,4]
##  [1,] 0.088119+0i -0.016260+0.011307i -0.016260-0.011307i -2.1834e-01+0i
##  [2,] 0.087237+0i -0.029344+0.004633i -0.029344-0.004633i  1.3986e-17+0i
##  [3,] 0.084447+0i  0.233302+0.000000i  0.233302+0.000000i  1.9105e-01+0i
##  [4,] 0.088853+0i -0.010392+0.010380i -0.010392-0.010380i  2.4895e-16+0i
##  [5,] 0.088119+0i -0.016260+0.011307i -0.016260-0.011307i -3.8989e-03+0i
##  [6,] 0.088853+0i -0.010392+0.010380i -0.010392-0.010380i -9.2309e-17+0i
##  [7,] 0.088853+0i -0.010392+0.010380i -0.010392-0.010380i -2.7972e-17+0i
##  [8,] 0.040388+0i  0.043881-0.035020i  0.043881+0.035020i -8.4267e-17+0i
##  [9,] 0.088119+0i -0.016260+0.011307i -0.016260-0.011307i -3.8989e-03+0i
## [10,] 0.088853+0i -0.010392+0.010380i -0.010392-0.010380i -2.7972e-17+0i
## [11,] 0.088853+0i -0.010392+0.010380i -0.010392-0.010380i  2.5175e-17+0i
## [12,] 0.079307+0i -0.147100-0.055435i -0.147100+0.055435i  3.5090e-02+0i
##                 [,5]           [,6]           [,7]           [,8]
##  [1,] -1.5214e-01+0i  8.8542e-02+0i -5.9653e-02+0i -4.5218e-17+0i
```

```
##  [2,]  2.9334e-17+0i  1.8933e-16+0i -5.9976e-16+0i -1.9170e-16+0i
##  [3,]  2.4006e-01+0i  2.0506e-01+0i -1.3952e-01+0i -1.8204e-15+0i
##  [4,]  4.1875e-16+0i  6.4080e-16+0i -6.8345e-16+0i -3.3548e-03+0i
##  [5,]  1.0207e-03+0i -8.8687e-02+0i -2.4622e-02+0i -1.2237e-16+0i
##  [6,] -2.7501e-16+0i -4.5633e-16+0i -2.4409e-17+0i  1.1983e-02+0i
##  [7,]  1.1734e-16+0i -6.3109e-17+0i  1.2902e-16+0i  1.4193e-01+0i
##  [8,] -8.9939e-17+0i -1.3471e-16+0i  2.1532e-16+0i  1.5628e-16+0i
##  [9,] -6.3172e-02+0i -4.4664e-02+0i -2.3669e-02+0i -4.5415e-16+0i
## [10,] -1.1330e-16+0i  1.7234e-16+0i  8.0201e-17+0i -2.3929e-01+0i
## [11,]  2.7501e-17+0i -5.5827e-17+0i  1.1681e-16+0i  8.8736e-02+0i
## [12,] -2.5767e-02+0i -1.6025e-01+0i  2.4747e-01+0i  2.6301e-15+0i
##                 [,9]          [,10]          [,11]          [,12]
##  [1,]  8.4464e-17+0i -2.3093e-16+0i -1.4747e-16+0i  1.8149e-17+0i
##  [2,] -8.6241e-17+0i  3.8210e-16+0i -7.1049e-18+0i  6.0498e-17+0i
##  [3,] -1.1941e-15+0i  5.6880e-15+0i  5.9192e-16+0i  2.0668e-01+0i
##  [4,] -5.1740e-02+0i  1.8961e-02+0i  2.5536e-01+0i -3.2265e-17+0i
##  [5,]  1.8855e-16+0i -7.6616e-16+0i -1.7778e-16+0i -1.0083e-17+0i
##  [6,] -1.3000e-01+0i  2.0275e-01+0i -1.0810e-01+0i  1.4116e-17+0i
##  [7,] -7.7672e-02+0i -1.9181e-01+0i -6.3841e-02+0i  1.4116e-17+0i
##  [8,]  1.2277e-16+0i -5.2261e-16+0i -2.9798e-17+0i -2.0668e-01+0i
##  [9,] -6.9005e-17+0i -4.3137e-17+0i -1.0274e-16+0i -1.0083e-17+0i
## [10,]  1.5310e-02+0i -7.2943e-02+0i -6.3841e-02+0i  1.4116e-17+0i
## [11,]  2.4410e-01+0i  4.3046e-02+0i -1.9582e-02+0i  1.0083e-17+0i
## [12,]  1.0646e-15+0i -4.8966e-15+0i -1.8985e-16+0i -4.1340e-17+0i
```

```
# Unique State Steady vector
W_ssv[,1]
```

```
##  [1] 0.088119+0i 0.087237+0i 0.084447+0i 0.088853+0i 0.088119+0i 0.088853+0i
##  [7] 0.088853+0i 0.040388+0i 0.088119+0i 0.088853+0i 0.088853+0i 0.079307+0i
```

4. IMDB reference

Adjacent matrix

```
##              SnowWhite Pinocchio Bambi LionKing Fantasia ToyStory Beauty&Beast
## SnowWhite            0         1     1        0        1        0            1
## Pinocchio            1         0     1        0        1        0            0
## Bambi                1         1     0        0        1        0            1
## LionKing             0         0     0        0        0        1            0
## Fantasia             0         1     1        0        0        0            0
## ToyStory             0         0     0        1        0        0            0
## Beauty&Beast         0         0     0        0        0        0            0
## Shrek                0         0     0        1        0        0            0
## Cinderella           1         1     1        0        1        0            1
## Nemo                 0         0     0        1        0        1            0
## WallE                0         0     0        1        0        1            0
## Up                   0         0     0        1        0        1            0
##              Shrek Cinderella Nemo WallE Up
## SnowWhite        0          1    0     0  0
## Pinocchio        0          1    0     0  0
## Bambi            0          1    0     0  0
## LionKing         1          0    1     1  1
## Fantasia         0          0    0     0  0
## ToyStory         1          0    1     1  1
```

```
## Beauty&Beast     0          1    0     1  0
## Shrek            0          0    0     0  0
## Cinderella       0          0    0     0  0
## Nemo             0          0    0     1  0
## WallE            0          0    1     0  1
## Up               0          0    1     1  0
```

Stochastic matrix

```
##              SnowWhite Pinocchio Bambi LionKing Fantasia ToyStory Beauty&Beast
## SnowWhite      0.00000      0.25  0.25      0.0     0.25     0.00      0.33333
## Pinocchio      0.33333      0.00  0.25      0.0     0.25     0.00      0.00000
## Bambi          0.33333      0.25  0.00      0.0     0.25     0.00      0.33333
## LionKing       0.00000      0.00  0.00      0.0     0.00     0.25      0.00000
## Fantasia       0.00000      0.25  0.25      0.0     0.00     0.00      0.00000
## ToyStory       0.00000      0.00  0.00      0.2     0.00     0.00      0.00000
## Beauty&Beast   0.00000      0.00  0.00      0.0     0.00     0.00      0.00000
## Shrek          0.00000      0.00  0.00      0.2     0.00     0.00      0.00000
## Cinderella     0.33333      0.25  0.25      0.0     0.25     0.00      0.33333
## Nemo           0.00000      0.00  0.00      0.2     0.00     0.25      0.00000
## WallE          0.00000      0.00  0.00      0.2     0.00     0.25      0.00000
## Up             0.00000      0.00  0.00      0.2     0.00     0.25      0.00000
##              Shrek Cinderella Nemo WallE      Up
## SnowWhite      0.0       0.25 0.00   0.0 0.00000
## Pinocchio      0.0       0.25 0.00   0.0 0.00000
## Bambi          0.0       0.25 0.00   0.0 0.00000
## LionKing       0.5       0.00 0.25   0.2 0.33333
## Fantasia       0.0       0.00 0.00   0.0 0.00000
## ToyStory       0.5       0.00 0.25   0.2 0.33333
## Beauty&Beast   0.0       0.25 0.00   0.2 0.00000
## Shrek          0.0       0.00 0.00   0.0 0.00000
## Cinderella     0.0       0.00 0.00   0.0 0.00000
## Nemo           0.0       0.00 0.00   0.2 0.00000
## WallE          0.0       0.00 0.25   0.0 0.33333
## Up             0.0       0.00 0.25   0.2 0.00000
```

Checking the stochastic matrix (All columns sum up to 1)

```
lapply(list(M), colSums, na.rm=TRUE)
```

```
## [[1]]
##    SnowWhite    Pinocchio        Bambi     LionKing     Fantasia     ToyStory
##            1            1            1            1            1            1
## Beauty&Beast        Shrek   Cinderella         Nemo        WallE           Up
##            1            1            1            1            1            1
```

Questions:

(a) Were there dangling nodes?

No. There weren't any columns with zero elements.

(b) Is the network connected?

Yes. The adjacent matrix and the transposed matrix were summed, and used to check the connectivity.

```
M_adj + t(M_adj)
```

```
##              SnowWhite Pinocchio Bambi LionKing Fantasia ToyStory Beauty&Beast
## SnowWhite            0         2     2        0        1        0            1
## Pinocchio            2         0     2        0        2        0            0
## Bambi                2         2     0        0        2        0            1
## LionKing             0         0     0        0        0        2            0
## Fantasia             1         2     2        0        0        0            0
## ToyStory             0         0     0        2        0        0            0
## Beauty&Beast         1         0     1        0        0        0            0
## Shrek                0         0     0        2        0        1            0
## Cinderella           2         2     2        0        1        0            2
## Nemo                 0         0     0        2        0        2            0
## WallE                0         0     0        2        0        2            1
## Up                   0         0     0        2        0        2            0
##              Shrek Cinderella Nemo WallE Up
## SnowWhite        0          2    0     0  0
## Pinocchio        0          2    0     0  0
## Bambi            0          2    0     0  0
## LionKing         2          0    2     2  2
## Fantasia         0          1    0     0  0
## ToyStory         1          0    2     2  2
## Beauty&Beast     0          2    0     1  0
## Shrek            0          0    0     0  0
## Cinderella       0          0    0     0  0
## Nemo             0          0    0     2  1
## WallE            0          0    2     0  2
## Up               0          0    1     2  0
```

(c) Is it regular?

Yes. After powering matrix W by 3000, we could verify that all the columns had similar values with similar patterns.

```
M %^% 3000
```

```
##              SnowWhite Pinocchio    Bambi   LionKing Fantasia   ToyStory
## SnowWhite     0.203620  0.203620 0.203620 2.0362e-01 0.203620 2.0362e-01
## Pinocchio     0.202715  0.202715 0.202715 2.0271e-01 0.202715 2.0271e-01
## Bambi         0.217195  0.217195 0.217195 2.1719e-01 0.217195 2.1719e-01
## LionKing      0.000000  0.000000 0.000000 1.1485e-54 0.000000 1.1309e-54
## Fantasia      0.104977  0.104977 0.104977 1.0498e-01 0.104977 1.0498e-01
## ToyStory      0.000000  0.000000 0.000000 1.1010e-54 0.000000 1.0842e-54
## Beauty&Beast  0.054299  0.054299 0.054299 5.4299e-02 0.054299 5.4299e-02
## Shrek         0.000000  0.000000 0.000000 2.3929e-55 0.000000 2.3563e-55
## Cinderella    0.217195  0.217195 0.217195 2.1719e-01 0.217195 2.1719e-01
## Nemo          0.000000  0.000000 0.000000 7.4383e-55 0.000000 7.3247e-55
## WallE         0.000000  0.000000 0.000000 1.0453e-54 0.000000 1.0294e-54
## Up            0.000000  0.000000 0.000000 9.3755e-55 0.000000 9.2323e-55
##              Beauty&Beast      Shrek Cinderella       Nemo      WallE
## SnowWhite        0.203620 2.0362e-01   0.203620 2.0362e-01 2.0362e-01
```

```
## Pinocchio        0.202715 2.0271e-01   0.202715 2.0271e-01 2.0271e-01
## Bambi            0.217195 2.1719e-01   0.217195 2.1719e-01 2.1719e-01
## LionKing         0.000000 1.1873e-54   0.000000 1.1309e-54 9.4375e-55
## Fantasia         0.104977 1.0498e-01   0.104977 1.0498e-01 1.0498e-01
## ToyStory         0.000000 1.1382e-54   0.000000 1.0842e-54 9.0475e-55
## Beauty&Beast     0.054299 5.4299e-02   0.054299 5.4299e-02 5.4299e-02
## Shrek            0.000000 2.4737e-55   0.000000 2.3563e-55 1.9663e-55
## Cinderella       0.217195 2.1719e-01   0.217195 2.1719e-01 2.1719e-01
## Nemo             0.000000 7.6897e-55   0.000000 7.3247e-55 6.1123e-55
## WallE            0.000000 1.0807e-54   0.000000 1.0294e-54 8.5899e-55
## Up               0.000000 9.6924e-55   0.000000 9.2323e-55 7.7042e-55
##                       Up
## SnowWhite    2.0362e-01
## Pinocchio    2.0271e-01
## Bambi        2.1719e-01
## LionKing     1.1192e-54
## Fantasia     1.0498e-01
## ToyStory     1.0730e-54
## Beauty&Beast 5.4299e-02
## Shrek        2.3320e-55
## Cinderella   2.1719e-01
## Nemo         7.2490e-55
## WallE        1.0187e-54
## Up           9.1369e-55
```

(d) Is there a unique steady state vector?

The eigenvectors corresponding to the eigenvalue 1 is given as below.

Eigenvalues

```
eigen(M)$values
```

```
##  [1]  1.000000+0.00000i  0.959915+0.00000i -0.408965+0.00000i -0.289641+0.06084i
##  [5] -0.289641-0.06084i -0.250000+0.00000i -0.250000+0.00000i -0.250000+0.00000i
##  [9] -0.200825+0.04276i -0.200825-0.04276i  0.100701+0.00000i  0.079281+0.00000i
```

Eigenvectors

```
M_eigenvector <- eigen(M)$vectors
M_eigenvector
```

```
##                [,1]         [,2]         [,3]                    [,4]
##  [1,] 4.6607e-01+0i -0.332439+0i -0.044185+0i -1.8217e-01+4.8301e-02i
##  [2,] 4.6400e-01+0i -0.348251+0i  0.085996+0i  7.0710e-01+0.0000e+00i
##  [3,] 4.9714e-01+0i -0.355336+0i -0.021022+0i -2.1590e-02+1.9321e-01i
##  [4,] 1.5443e-17+0i  0.352398+0i  0.458918+0i -1.3744e-16+1.7355e-16i
##  [5,] 2.4029e-01+0i -0.183242+0i -0.039718+0i -5.3314e-01-2.7874e-01i
##  [6,] 1.3042e-17+0i  0.337835+0i  0.603264+0i -9.4106e-16-4.6291e-16i
##  [7,] 1.2429e-01+0i -0.025715+0i  0.051036+0i  5.1395e-02-1.5597e-01i
##  [8,] 3.1264e-18+0i  0.073423+0i -0.224429+0i  1.0416e-16-1.0319e-16i
##  [9,] 4.9714e-01+0i -0.355336+0i -0.021022+0i -2.1590e-02+1.9321e-01i
## [10,] 8.8432e-18+0i  0.228237+0i -0.555018+0i  8.0049e-16+5.5077e-16i
## [11,] 1.2282e-17+0i  0.320748+0i -0.078083+0i -3.6214e-17-1.6719e-16i
```

```
## [12,] 1.1054e-17+0i  0.287679+0i -0.215736+0i  2.3337e-16-4.5829e-17i
##                          [,5]           [,6]           [,7]           [,8]
##  [1,] -1.8217e-01-4.8301e-02i -2.2461e-15+0i  5.7012e-17+0i  3.6234e-16+0i
##  [2,]  7.0710e-01+0.0000e+00i  1.8451e-01+0i  3.8639e-01+0i -6.1086e-01+0i
##  [3,] -2.1590e-02-1.9321e-01i  5.9389e-01+0i -4.0631e-01+0i -1.2676e-01+0i
##  [4,] -1.3744e-16-1.7355e-16i  1.8227e-16+0i -7.9817e-16+0i -7.3052e-17+0i
##  [5,] -5.3314e-01+2.7874e-01i -7.7840e-01+0i  1.9917e-02+0i  7.3762e-01+0i
##  [6,] -9.4106e-16+4.6291e-16i -6.0546e-02+0i -5.8533e-01+0i -1.8261e-01+0i
##  [7,]  5.1395e-02+1.5597e-01i -9.5989e-16+0i  3.7414e-16+0i  1.4771e-15+0i
##  [8,]  1.0416e-16+1.0319e-16i -6.9616e-17+0i  4.2046e-16+0i  1.3697e-17+0i
##  [9,] -2.1590e-02-1.9321e-01i  1.4176e-15+0i -1.1402e-16+0i -2.5247e-15+0i
## [10,]  8.0049e-16-5.5077e-16i  6.0546e-02+0i  5.8533e-01+0i  1.8261e-01+0i
## [11,] -3.6214e-17+1.6719e-16i -4.2761e-16+0i  1.8529e-16+0i  9.5260e-16+0i
## [12,]  2.3337e-16+4.5829e-17i  1.5695e-16+0i -3.4207e-16+0i -4.2078e-16+0i
##                       [,9]               [,10]         [,11]           [,12]
##  [1,]  0.065572+0.066706i  0.065572-0.066706i  0.122867+0i -1.4215e-01+0i
##  [2,]  0.433594-0.207732i  0.433594+0.207732i -0.406510+0i  3.9047e-01+0i
##  [3,]  0.184818+0.076049i  0.184818-0.076049i  0.152062+0i -1.7812e-01+0i
##  [4,]  0.006262-0.016055i  0.006262+0.016055i -0.045961+0i  1.5297e-16+0i
##  [5,] -0.769839+0.000000i -0.769839+0.000000i -0.631690+0i  6.6961e-01+0i
##  [6,]  0.010720-0.003607i  0.010720+0.003607i -0.039408+0i  1.5365e-16+0i
##  [7,] -0.073107+0.009949i -0.073107-0.009949i  0.587675+0i -5.6169e-01+0i
##  [8,] -0.009222+0.014025i -0.009222-0.014025i -0.091282+0i  2.8282e-16+0i
##  [9,]  0.184818+0.076049i  0.184818-0.076049i  0.152062+0i -1.7812e-01+0i
## [10,]  0.104799+0.162982i  0.104799-0.162982i  0.021051+0i -5.2992e-17+0i
## [11,] -0.159741-0.120683i -0.159741+0.120683i  0.105821+0i -3.2517e-16+0i
## [12,]  0.021326-0.057683i  0.021326+0.057683i  0.073313+0i -2.2010e-16+0i
```

Checking the eigenvectors (Sum up to 1)

```
M_ssv <- M_eigenvector / sum(M_eigenvector)
M_ssv
```

```
##                [,1]          [,2]           [,3]                     [,4]
##  [1,] 2.0362e-01+0i -0.145238+0i -0.0193040+0i -7.9588e-02+2.1102e-02i
##  [2,] 2.0271e-01+0i -0.152146+0i  0.0375703+0i  3.0892e-01+0.0000e+00i
##  [3,] 2.1719e-01+0i -0.155242+0i -0.0091844+0i -9.4324e-03+8.4409e-02i
##  [4,] 6.7469e-18+0i  0.153958+0i  0.2004954+0i -6.0048e-17+7.5823e-17i
##  [5,] 1.0498e-01+0i -0.080056+0i -0.0173523+0i -2.3292e-01-1.2178e-01i
##  [6,] 5.6977e-18+0i  0.147596+0i  0.2635580+0i -4.1114e-16-2.0224e-16i
##  [7,] 5.4299e-02+0i -0.011235+0i  0.0222971+0i  2.2454e-02-6.8140e-02i
##  [8,] 1.3659e-18+0i  0.032077+0i -0.0980501+0i  4.5508e-17-4.5082e-17i
##  [9,] 2.1719e-01+0i -0.155242+0i -0.0091844+0i -9.4324e-03+8.4409e-02i
## [10,] 3.8635e-18+0i  0.099714+0i -0.2424801+0i  3.4972e-16+2.4062e-16i
## [11,] 5.3660e-18+0i  0.140131+0i -0.0341132+0i -1.5821e-17-7.3042e-17i
## [12,] 4.8294e-18+0i  0.125683+0i -0.0942523+0i  1.0196e-16-2.0022e-17i
##                          [,5]           [,6]           [,7]           [,8]
##  [1,] -7.9588e-02-2.1102e-02i -9.8130e-16+0i  2.4908e-17+0i  1.5830e-16+0i
##  [2,]  3.0892e-01+0.0000e+00i  8.0610e-02+0i  1.6881e-01+0i -2.6688e-01+0i
##  [3,] -9.4324e-03-8.4409e-02i  2.5946e-01+0i -1.7751e-01+0i -5.5380e-02+0i
##  [4,] -6.0048e-17-7.5823e-17i  7.9630e-17+0i -3.4871e-16+0i -3.1916e-17+0i
##  [5,] -2.3292e-01+1.2178e-01i -3.4007e-01+0i  8.7015e-03+0i  3.2226e-01+0i
##  [6,] -4.1114e-16+2.0224e-16i -2.6452e-02+0i -2.5572e-01+0i -7.9780e-02+0i
##  [7,]  2.2454e-02+6.8140e-02i -4.1936e-16+0i  1.6346e-16+0i  6.4533e-16+0i
```

```
##  [8,]  4.5508e-17+4.5082e-17i -3.0414e-17+0i  1.8369e-16+0i  5.9842e-18+0i
##  [9,] -9.4324e-03-8.4409e-02i  6.1934e-16+0i -4.9816e-17+0i -1.1030e-15+0i
## [10,]  3.4972e-16-2.4062e-16i  2.6452e-02+0i  2.5572e-01+0i  7.9780e-02+0i
## [11,] -1.5821e-17+7.3042e-17i -1.8682e-16+0i  8.0950e-17+0i  4.1618e-16+0i
## [12,]  1.0196e-16+2.0022e-17i  6.8570e-17+0i -1.4945e-16+0i -1.8383e-16+0i
##                        [,9]                 [,10]        [,11]          [,12]
##  [1,]  0.0286475+0.0291428i  0.0286475-0.0291428i  0.053679+0i -6.2103e-02+0i
##  [2,]  0.1894315-0.0907555i  0.1894315+0.0907555i -0.177599+0i  1.7059e-01+0i
##  [3,]  0.0807445+0.0332249i  0.0807445-0.0332249i  0.066434+0i -7.7820e-02+0i
##  [4,]  0.0027357-0.0070141i  0.0027357+0.0070141i -0.020080+0i  6.6831e-17+0i
##  [5,] -0.3363322+0.0000000i -0.3363322+0.0000000i -0.275977+0i  2.9254e-01+0i
##  [6,]  0.0046832-0.0015759i  0.0046832+0.0015759i -0.017217+0i  6.7126e-17+0i
##  [7,] -0.0319394+0.0043467i -0.0319394-0.0043467i  0.256747+0i -2.4539e-01+0i
##  [8,] -0.0040292+0.0061273i -0.0040292-0.0061273i -0.039880+0i  1.2356e-16+0i
##  [9,]  0.0807445+0.0332249i  0.0807445-0.0332249i  0.066434+0i -7.7820e-02+0i
## [10,]  0.0457855+0.0712048i  0.0457855-0.0712048i  0.009197+0i -2.3152e-17+0i
## [11,] -0.0697889-0.0527249i -0.0697889+0.0527249i  0.046232+0i -1.4206e-16+0i
## [12,]  0.0093172-0.0252009i  0.0093172+0.0252009i  0.032029+0i -9.6157e-17+0i
```

```
# Unique state steady vector
M_ssv[,1]
```

```
##  [1] 2.0362e-01+0i 2.0271e-01+0i 2.1719e-01+0i 6.7469e-18+0i 1.0498e-01+0i
##  [6] 5.6977e-18+0i 5.4299e-02+0i 1.3659e-18+0i 2.1719e-01+0i 3.8635e-18+0i
## [11] 5.3660e-18+0i 4.8294e-18+0i
```

5. Average of the matrices W and M

```
A = (W+M)/2
A
```

```
##              SnowWhite Pinocchio Bambi LionKing Fantasia ToyStory Beauty&Beast
## SnowWhite      0.00000  0.180556 0.175 0.045455    0.175 0.045455     0.212121
## Pinocchio      0.21667  0.000000 0.175 0.045455    0.175 0.045455     0.045455
## Bambi          0.21667  0.180556 0.000 0.045455    0.175 0.045455     0.212121
## LionKing       0.05000  0.055556 0.050 0.000000    0.050 0.170455     0.045455
## Fantasia       0.05000  0.180556 0.175 0.045455    0.000 0.045455     0.045455
## ToyStory       0.05000  0.055556 0.050 0.145455    0.050 0.000000     0.045455
## Beauty&Beast   0.05000  0.055556 0.050 0.045455    0.050 0.045455     0.000000
## Shrek          0.00000  0.000000 0.000 0.145455    0.000 0.045455     0.045455
## Cinderella     0.21667  0.180556 0.175 0.045455    0.175 0.045455     0.212121
## Nemo           0.05000  0.055556 0.050 0.145455    0.050 0.170455     0.045455
## WallE          0.05000  0.055556 0.050 0.145455    0.050 0.170455     0.045455
## Up             0.05000  0.000000 0.050 0.145455    0.050 0.170455     0.045455
##               Shrek Cinderella     Nemo     WallE      Up
## SnowWhite      0.05      0.175 0.045455 0.045455 0.05000
## Pinocchio      0.05      0.175 0.045455 0.045455 0.05000
## Bambi          0.00      0.175 0.045455 0.045455 0.05000
## LionKing       0.30      0.050 0.170455 0.145455 0.21667
## Fantasia       0.05      0.050 0.045455 0.045455 0.05000
## ToyStory       0.30      0.050 0.170455 0.145455 0.21667
## Beauty&Beast   0.05      0.175 0.045455 0.145455 0.05000
## Shrek          0.00      0.000 0.045455 0.045455 0.00000
## Cinderella     0.05      0.000 0.045455 0.045455 0.05000
```

```
## Nemo          0.05      0.050 0.000000 0.145455 0.05000
## WallE         0.05      0.050 0.170455 0.000000 0.21667
## Up            0.05      0.050 0.170455 0.145455 0.00000
```

Checking the matrix A (All columns sum up to 1)

```
lapply(list(A), colSums, na.rm=TRUE)
```

```
## [[1]]
##    SnowWhite    Pinocchio        Bambi     LionKing     Fantasia     ToyStory
##            1            1            1            1            1            1
## Beauty&Beast        Shrek   Cinderella        Nemo        WallE           Up
##            1            1            1            1            1            1
```

(a) Were there dangling nodes?

No. There weren't any columns with zero elements.

(b) Is the network connected?

Since two connected matrices were averaged, the network must be connected.

(c) Is it regular?

Yes. The Average matrix is regular. Matrix was powered to the 3000 to check the regularity.

```
A %^% 3000
```

```
##              SnowWhite Pinocchio    Bambi LionKing Fantasia ToyStory
## SnowWhite     0.101611  0.101611 0.101611 0.101611 0.101611 0.101611
## Pinocchio     0.095187  0.095187 0.095187 0.095187 0.095187 0.095187
## Bambi         0.104007  0.104007 0.104007 0.104007 0.104007 0.104007
## LionKing      0.094242  0.094242 0.094242 0.094242 0.094242 0.094242
## Fantasia      0.070031  0.070031 0.070031 0.070031 0.070031 0.070031
## ToyStory      0.092229  0.092229 0.092229 0.092229 0.092229 0.092229
## Beauty&Beast  0.067521  0.067521 0.067521 0.067521 0.067521 0.067521
## Shrek         0.028380  0.028380 0.028380 0.028380 0.028380 0.028380
## Cinderella    0.105214  0.105214 0.105214 0.105214 0.105214 0.105214
## Nemo          0.074983  0.074983 0.074983 0.074983 0.074983 0.074983
## WallE         0.088048  0.088048 0.088048 0.088048 0.088048 0.088048
## Up            0.078548  0.078548 0.078548 0.078548 0.078548 0.078548
##              Beauty&Beast    Shrek Cinderella     Nemo    WallE       Up
## SnowWhite        0.101611 0.101611   0.101611 0.101611 0.101611 0.101611
## Pinocchio        0.095187 0.095187   0.095187 0.095187 0.095187 0.095187
## Bambi            0.104007 0.104007   0.104007 0.104007 0.104007 0.104007
## LionKing         0.094242 0.094242   0.094242 0.094242 0.094242 0.094242
## Fantasia         0.070031 0.070031   0.070031 0.070031 0.070031 0.070031
## ToyStory         0.092229 0.092229   0.092229 0.092229 0.092229 0.092229
## Beauty&Beast     0.067521 0.067521   0.067521 0.067521 0.067521 0.067521
## Shrek            0.028380 0.028380   0.028380 0.028380 0.028380 0.028380
## Cinderella       0.105214 0.105214   0.105214 0.105214 0.105214 0.105214
## Nemo             0.074983 0.074983   0.074983 0.074983 0.074983 0.074983
## WallE            0.088048 0.088048   0.088048 0.088048 0.088048 0.088048
## Up               0.078548 0.078548   0.078548 0.078548 0.078548 0.078548
```

(d) Is there a unique steady state vector?

Eigenvalues

```
eigen(A)$values
```

```
##  [1]  1.000000+0.000000i  0.463343+0.000000i -0.257073+0.000000i
##  [4] -0.237940+0.000000i -0.195389+0.000000i -0.175000+0.000000i
##  [7] -0.170455+0.000000i -0.156609+0.059083i -0.156609-0.059083i
## [10] -0.122779+0.000000i  0.004256+0.032844i  0.004256-0.032844i
```

Eigenvectors

```
A_eigenvector <- eigen(A)$vectors
A_eigenvector
```

```
##              [,1]         [,2]         [,3]         [,4]           [,5]
##  [1,] 0.341612+0i -0.334226+0i  0.047635+0i -0.154099+0i -0.30548460+0i
##  [2,] 0.320014+0i -0.340948+0i -0.164053+0i  0.423459+0i  0.33450291+0i
##  [3,] 0.349666+0i -0.370693+0i  0.186942+0i -0.239113+0i -0.00076925+0i
##  [4,] 0.316838+0i  0.355598+0i -0.460399+0i  0.377725+0i  0.24966771+0i
##  [5,] 0.235441+0i -0.186931+0i -0.020498+0i -0.125419+0i -0.29937240+0i
##  [6,] 0.310071+0i  0.341572+0i -0.593280+0i  0.517653+0i  0.49998684+0i
##  [7,] 0.227002+0i -0.046445+0i -0.086866+0i  0.165469+0i -0.00922715+0i
##  [8,] 0.095412+0i  0.187046+0i  0.268364+0i -0.235432+0i -0.13031071+0i
##  [9,] 0.353726+0i -0.356042+0i  0.023452+0i -0.052085+0i  0.31878446+0i
## [10,] 0.252088+0i  0.194833+0i  0.494990+0i -0.400754+0i -0.32681202+0i
## [11,] 0.296014+0i  0.278789+0i  0.140675+0i -0.258678+0i -0.40273477+0i
## [12,] 0.264076+0i  0.277448+0i  0.163040+0i -0.018727+0i  0.07176899+0i
##                 [,6]           [,7]                [,8]                [,9]
##  [1,]  1.4855e-15+0i  9.1658e-16+0i  0.075069+0.151918i  0.075069-0.151918i
##  [2,] -4.6336e-16+0i -3.0885e-16+0i  0.286714-0.320006i  0.286714+0.320006i
##  [3,] -7.0711e-01+0i  7.7710e-15+0i  0.165151+0.064414i  0.165151-0.064414i
##  [4,] -1.3492e-15+0i -1.1158e-15+0i  0.195687-0.032953i  0.195687+0.032953i
##  [5,]  7.0711e-01+0i -7.9968e-15+0i -0.555144+0.000000i -0.555144+0.000000i
##  [6,] -2.2360e-14+0i  7.0711e-01+0i  0.190512+0.048636i  0.190512-0.048636i
##  [7,]  3.2367e-16+0i  1.8929e-16+0i -0.181440+0.025693i -0.181440-0.025693i
##  [8,]  7.2741e-16+0i  7.3061e-16+0i -0.073976+0.052338i -0.073976-0.052338i
##  [9,] -4.4156e-15+0i -7.4389e-16+0i  0.187765+0.134056i  0.187765-0.134056i
## [10,]  1.9637e-14+0i -7.0711e-01+0i -0.373214+0.049463i -0.373214-0.049463i
## [11,]  4.6336e-15+0i  9.5643e-16+0i -0.075211-0.294823i -0.075211+0.294823i
## [12,] -6.6523e-16+0i -5.5626e-16+0i  0.158087+0.121265i  0.158087-0.121265i
##                [,10]               [,11]               [,12]
##  [1,] -1.2141e-01+0i -0.07857+0.080742i -0.07857-0.080742i
##  [2,] -1.9744e-01+0i  0.34348+0.021512i  0.34348-0.021512i
##  [3,] -2.0077e-01+0i -0.13533+0.029281i -0.13533-0.029281i
##  [4,]  7.6732e-02+0i  0.09834+0.124326i  0.09834-0.124326i
##  [5,]  6.9155e-01+0i  0.46133-0.145708i  0.46133+0.145708i
##  [6,]  3.6495e-02+0i  0.08152+0.109698i  0.08152-0.109698i
##  [7,]  5.0102e-05+0i -0.49670+0.000000i -0.49670+0.000000i
##  [8,] -1.8287e-02+0i  0.10422+0.289062i  0.10422-0.289062i
##  [9,] -2.1828e-01+0i -0.09291+0.102137i -0.09291-0.102137i
## [10,] -5.1105e-01+0i  0.07012-0.105387i  0.07012+0.105387i
## [11,]  2.7836e-01+0i -0.16874-0.299783i -0.16874+0.299783i
```

```
## [12,]  1.8406e-01+0i -0.18676-0.205881i -0.18676+0.205881i
```

```
# Unique state steady vector
A_ssv <- A_eigenvector / sum(A_eigenvector)
A_ssv[,1]
```

```
##  [1] 0.101611+0i 0.095187+0i 0.104007+0i 0.094242+0i 0.070031+0i 0.092229+0i
##  [7] 0.067521+0i 0.028380+0i 0.105214+0i 0.074983+0i 0.088048+0i 0.078548+0i
```

6. Page-rank: Damping factor (85%)

Matrix A * 85%

```
##              SnowWhite Pinocchio   Bambi LionKing Fantasia ToyStory
## SnowWhite      0.00000  0.153472 0.14875 0.038636  0.14875 0.038636
## Pinocchio      0.18417  0.000000 0.14875 0.038636  0.14875 0.038636
## Bambi          0.18417  0.153472 0.00000 0.038636  0.14875 0.038636
## LionKing       0.04250  0.047222 0.04250 0.000000  0.04250 0.144886
## Fantasia       0.04250  0.153472 0.14875 0.038636  0.00000 0.038636
## ToyStory       0.04250  0.047222 0.04250 0.123636  0.04250 0.000000
## Beauty&Beast   0.04250  0.047222 0.04250 0.038636  0.04250 0.038636
## Shrek          0.00000  0.000000 0.00000 0.123636  0.00000 0.038636
## Cinderella     0.18417  0.153472 0.14875 0.038636  0.14875 0.038636
## Nemo           0.04250  0.047222 0.04250 0.123636  0.04250 0.144886
## WallE          0.04250  0.047222 0.04250 0.123636  0.04250 0.144886
## Up             0.04250  0.000000 0.04250 0.123636  0.04250 0.144886
##              Beauty&Beast  Shrek Cinderella     Nemo    WallE      Up
## SnowWhite        0.180303 0.0425    0.14875 0.038636 0.038636 0.04250
## Pinocchio        0.038636 0.0425    0.14875 0.038636 0.038636 0.04250
## Bambi            0.180303 0.0000    0.14875 0.038636 0.038636 0.04250
## LionKing         0.038636 0.2550    0.04250 0.144886 0.123636 0.18417
## Fantasia         0.038636 0.0425    0.04250 0.038636 0.038636 0.04250
## ToyStory         0.038636 0.2550    0.04250 0.144886 0.123636 0.18417
## Beauty&Beast     0.000000 0.0425    0.14875 0.038636 0.123636 0.04250
## Shrek            0.038636 0.0000    0.00000 0.038636 0.038636 0.00000
## Cinderella       0.180303 0.0425    0.00000 0.038636 0.038636 0.04250
## Nemo             0.038636 0.0425    0.04250 0.000000 0.123636 0.04250
## WallE            0.038636 0.0425    0.04250 0.144886 0.000000 0.18417
## Up               0.038636 0.0425    0.04250 0.144886 0.123636 0.00000
```

Matrix B (12 rows and columns of value 1/12)

```
##           [,1]     [,2]     [,3]     [,4]     [,5]     [,6]     [,7]     [,8]
##  [1,] 0.083333 0.083333 0.083333 0.083333 0.083333 0.083333 0.083333 0.083333
##  [2,] 0.083333 0.083333 0.083333 0.083333 0.083333 0.083333 0.083333 0.083333
##  [3,] 0.083333 0.083333 0.083333 0.083333 0.083333 0.083333 0.083333 0.083333
##  [4,] 0.083333 0.083333 0.083333 0.083333 0.083333 0.083333 0.083333 0.083333
##  [5,] 0.083333 0.083333 0.083333 0.083333 0.083333 0.083333 0.083333 0.083333
##  [6,] 0.083333 0.083333 0.083333 0.083333 0.083333 0.083333 0.083333 0.083333
##  [7,] 0.083333 0.083333 0.083333 0.083333 0.083333 0.083333 0.083333 0.083333
##  [8,] 0.083333 0.083333 0.083333 0.083333 0.083333 0.083333 0.083333 0.083333
##  [9,] 0.083333 0.083333 0.083333 0.083333 0.083333 0.083333 0.083333 0.083333
## [10,] 0.083333 0.083333 0.083333 0.083333 0.083333 0.083333 0.083333 0.083333
## [11,] 0.083333 0.083333 0.083333 0.083333 0.083333 0.083333 0.083333 0.083333
## [12,] 0.083333 0.083333 0.083333 0.083333 0.083333 0.083333 0.083333 0.083333
```

```
##           [,9]    [,10]    [,11]    [,12]
##  [1,] 0.083333 0.083333 0.083333 0.083333
##  [2,] 0.083333 0.083333 0.083333 0.083333
##  [3,] 0.083333 0.083333 0.083333 0.083333
##  [4,] 0.083333 0.083333 0.083333 0.083333
##  [5,] 0.083333 0.083333 0.083333 0.083333
##  [6,] 0.083333 0.083333 0.083333 0.083333
##  [7,] 0.083333 0.083333 0.083333 0.083333
##  [8,] 0.083333 0.083333 0.083333 0.083333
##  [9,] 0.083333 0.083333 0.083333 0.083333
## [10,] 0.083333 0.083333 0.083333 0.083333
## [11,] 0.083333 0.083333 0.083333 0.083333
## [12,] 0.083333 0.083333 0.083333 0.083333
```

Matrix F

```
F <- .85*A + .15*B
F
```

```
##              SnowWhite Pinocchio   Bambi LionKing Fantasia ToyStory
## SnowWhite      0.01250  0.165972 0.16125 0.051136  0.16125 0.051136
## Pinocchio      0.19667  0.012500 0.16125 0.051136  0.16125 0.051136
## Bambi          0.19667  0.165972 0.01250 0.051136  0.16125 0.051136
## LionKing       0.05500  0.059722 0.05500 0.012500  0.05500 0.157386
## Fantasia       0.05500  0.165972 0.16125 0.051136  0.01250 0.051136
## ToyStory       0.05500  0.059722 0.05500 0.136136  0.05500 0.012500
## Beauty&Beast   0.05500  0.059722 0.05500 0.051136  0.05500 0.051136
## Shrek          0.01250  0.012500 0.01250 0.136136  0.01250 0.051136
## Cinderella     0.19667  0.165972 0.16125 0.051136  0.16125 0.051136
## Nemo           0.05500  0.059722 0.05500 0.136136  0.05500 0.157386
## WallE          0.05500  0.059722 0.05500 0.136136  0.05500 0.157386
## Up             0.05500  0.012500 0.05500 0.136136  0.05500 0.157386
##              Beauty&Beast  Shrek Cinderella     Nemo    WallE      Up
## SnowWhite        0.192803 0.0550    0.16125 0.051136 0.051136 0.05500
## Pinocchio        0.051136 0.0550    0.16125 0.051136 0.051136 0.05500
## Bambi            0.192803 0.0125    0.16125 0.051136 0.051136 0.05500
## LionKing         0.051136 0.2675    0.05500 0.157386 0.136136 0.19667
## Fantasia         0.051136 0.0550    0.05500 0.051136 0.051136 0.05500
## ToyStory         0.051136 0.2675    0.05500 0.157386 0.136136 0.19667
## Beauty&Beast     0.012500 0.0550    0.16125 0.051136 0.136136 0.05500
## Shrek            0.051136 0.0125    0.01250 0.051136 0.051136 0.01250
## Cinderella       0.192803 0.0550    0.01250 0.051136 0.051136 0.05500
## Nemo             0.051136 0.0550    0.05500 0.012500 0.136136 0.05500
## WallE            0.051136 0.0550    0.05500 0.157386 0.012500 0.19667
## Up               0.051136 0.0550    0.05500 0.157386 0.136136 0.01250
```

Checking matrix F

```
lapply(list(F), colSums, na.rm=TRUE)
```

```
## [[1]]
##    SnowWhite    Pinocchio        Bambi     LionKing     Fantasia     ToyStory
##            1            1            1            1            1            1
## Beauty&Beast        Shrek   Cinderella         Nemo        WallE           Up
```

```
##            1            1            1            1            1            1
```

Checking the regularity of matrix F

Matrix F is regular.

```
F %^% 3000
```

```
##              SnowWhite Pinocchio    Bambi LionKing Fantasia ToyStory
## SnowWhite     0.098196  0.098196 0.098196 0.098196 0.098196 0.098196
## Pinocchio     0.092293  0.092293 0.092293 0.092293 0.092293 0.092293
## Bambi         0.099862  0.099862 0.099862 0.099862 0.099862 0.099862
## LionKing      0.094670  0.094670 0.094670 0.094670 0.094670 0.094670
## Fantasia      0.071200  0.071200 0.071200 0.071200 0.071200 0.071200
## ToyStory      0.092913  0.092913 0.092913 0.092913 0.092913 0.092913
## Beauty&Beast  0.069341  0.069341 0.069341 0.069341 0.069341 0.069341
## Shrek         0.036808  0.036808 0.036808 0.036808 0.036808 0.036808
## Cinderella    0.101224  0.101224 0.101224 0.101224 0.101224 0.101224
## Nemo          0.076238  0.076238 0.076238 0.076238 0.076238 0.076238
## WallE         0.087709  0.087709 0.087709 0.087709 0.087709 0.087709
## Up            0.079545  0.079545 0.079545 0.079545 0.079545 0.079545
##               Beauty&Beast    Shrek Cinderella     Nemo    WallE       Up
## SnowWhite         0.098196 0.098196   0.098196 0.098196 0.098196 0.098196
## Pinocchio         0.092293 0.092293   0.092293 0.092293 0.092293 0.092293
## Bambi             0.099862 0.099862   0.099862 0.099862 0.099862 0.099862
## LionKing          0.094670 0.094670   0.094670 0.094670 0.094670 0.094670
## Fantasia          0.071200 0.071200   0.071200 0.071200 0.071200 0.071200
## ToyStory          0.092913 0.092913   0.092913 0.092913 0.092913 0.092913
## Beauty&Beast      0.069341 0.069341   0.069341 0.069341 0.069341 0.069341
## Shrek             0.036808 0.036808   0.036808 0.036808 0.036808 0.036808
## Cinderella        0.101224 0.101224   0.101224 0.101224 0.101224 0.101224
## Nemo              0.076238 0.076238   0.076238 0.076238 0.076238 0.076238
## WallE             0.087709 0.087709   0.087709 0.087709 0.087709 0.087709
## Up                0.079545 0.079545   0.079545 0.079545 0.079545 0.079545
```

Film-rank

```
eigen(F)$values
```

Eigenvalues

```
##  [1]  1.000000+0.000000i  0.393841+0.000000i -0.218512+0.000000i
##  [4] -0.202249+0.000000i -0.166081+0.000000i -0.148750+0.000000i
##  [7] -0.144886+0.000000i -0.133118+0.050221i -0.133118-0.050221i
## [10] -0.104362+0.000000i  0.003618+0.027918i  0.003618-0.027918i
```

```
F_eigenvector <- eigen(F)$vectors
F_eigenvector
```

Film-rank (Steady-state vector)

```
##             [,1]         [,2]         [,3]         [,4]           [,5]
##  [1,] 0.33284+0i -0.334226+0i  0.047635+0i -0.154099+0i  0.30548460+0i
##  [2,] 0.31283+0i -0.340948+0i -0.164053+0i  0.423459+0i -0.33450291+0i
##  [3,] 0.33849+0i -0.370693+0i  0.186942+0i -0.239113+0i  0.00076925+0i
```

```
##  [4,] 0.32089+0i  0.355598+0i -0.460399+0i  0.377725+0i -0.24966771+0i
##  [5,] 0.24134+0i -0.186931+0i -0.020498+0i -0.125419+0i  0.29937240+0i
##  [6,] 0.31493+0i  0.341572+0i -0.593280+0i  0.517653+0i -0.49998684+0i
##  [7,] 0.23503+0i -0.046445+0i -0.086866+0i  0.165469+0i  0.00922715+0i
##  [8,] 0.12476+0i  0.187046+0i  0.268364+0i -0.235432+0i  0.13031071+0i
##  [9,] 0.34310+0i -0.356042+0i  0.023452+0i -0.052085+0i -0.31878446+0i
## [10,] 0.25841+0i  0.194833+0i  0.494990+0i -0.400754+0i  0.32681202+0i
## [11,] 0.29729+0i  0.278789+0i  0.140675+0i -0.258678+0i  0.40273477+0i
## [12,] 0.26962+0i  0.277448+0i  0.163040+0i -0.018727+0i -0.07176899+0i
##                 [,6]           [,7]                [,8]                [,9]
##  [1,] -2.2078e-15+0i  6.3760e-16+0i -0.075069-0.151918i -0.075069+0.151918i
##  [2,]  3.6268e-15+0i  2.6567e-17+0i -0.286714+0.320006i -0.286714-0.320006i
##  [3,] -7.0711e-01+0i  3.6529e-15+0i -0.165151-0.064414i -0.165151+0.064414i
##  [4,]  2.7393e-15+0i -5.3134e-17+0i -0.195687+0.032953i -0.195687-0.032953i
##  [5,]  7.0711e-01+0i -4.1345e-15+0i  0.555144+0.000000i  0.555144+0.000000i
##  [6,]  1.9041e-14+0i  7.0711e-01+0i -0.190512-0.048636i -0.190512+0.048636i
##  [7,] -4.7699e-16+0i  5.0809e-16+0i  0.181440-0.025693i  0.181440+0.025693i
##  [8,] -1.3219e-15+0i  1.1955e-16+0i  0.073976-0.052338i  0.073976+0.052338i
##  [9,]  2.9982e-15+0i -1.2354e-15+0i -0.187765-0.134056i -0.187765+0.134056i
## [10,] -1.7283e-14+0i -7.0711e-01+0i  0.373214-0.049463i  0.373214+0.049463i
## [11,] -4.7018e-15+0i  1.9327e-15+0i  0.075211+0.294823i  0.075211-0.294823i
## [12,]  7.8874e-16+0i  4.6326e-16+0i -0.158087-0.121265i -0.158087+0.121265i
##                [,10]              [,11]              [,12]
##  [1,] -1.2141e-01+0i  0.07857-0.080742i  0.07857+0.080742i
##  [2,] -1.9744e-01+0i -0.34348-0.021512i -0.34348+0.021512i
##  [3,] -2.0077e-01+0i  0.13533-0.029281i  0.13533+0.029281i
##  [4,]  7.6732e-02+0i -0.09834-0.124326i -0.09834+0.124326i
##  [5,]  6.9155e-01+0i -0.46133+0.145708i -0.46133-0.145708i
##  [6,]  3.6495e-02+0i -0.08152-0.109698i -0.08152+0.109698i
##  [7,]  5.0102e-05+0i  0.49670+0.000000i  0.49670+0.000000i
##  [8,] -1.8287e-02+0i -0.10422-0.289062i -0.10422+0.289062i
##  [9,] -2.1828e-01+0i  0.09291-0.102137i  0.09291+0.102137i
## [10,] -5.1105e-01+0i -0.07012+0.105387i -0.07012-0.105387i
## [11,]  2.7836e-01+0i  0.16874+0.299783i  0.16874-0.299783i
## [12,]  1.8406e-01+0i  0.18676+0.205881i  0.18676-0.205881i
```

```
# Unique state steady vector
F_ssv <- F_eigenvector / sum(F_eigenvector)
FilmRank <- F_ssv[,1]
```

Film Rank is,

```
FilmRank
```

```
##  [1] 0.098196+0i 0.092293+0i 0.099862+0i 0.094670+0i 0.071200+0i 0.092913+0i
##  [7] 0.069341+0i 0.036808+0i 0.101224+0i 0.076238+0i 0.087709+0i 0.079545+0i
```

7. Comparison between three lists (AFI, Group, Film-rank)

Is there any correlation?

Could you use the film-rank to spot which two did not appear in the original top ten list?

If you have any ideas on how to "improve" film-rank which are easy to implement, you're welcome to give it a try!

Comparison between AFI, Group list, and Film Rank

```
FR <- data.frame(AFI, FilmRank)
FilmRankList <- FR[order(FR$FilmRank, decreasing = TRUE),]

Final <- data.frame(AFI, GroupList, FilmRankList[1])

names(Final)[3] <- "FilmRank"
row.names(Final) <- NULL
Final
```

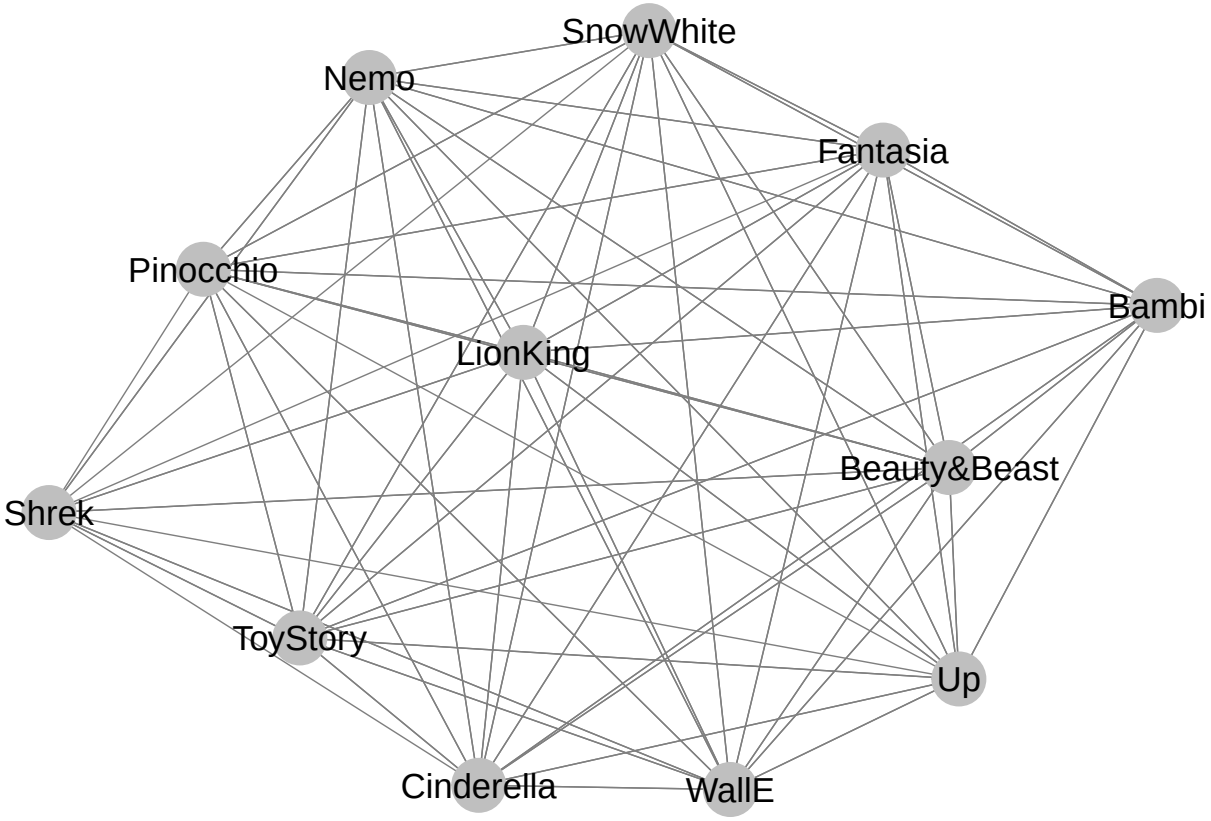
```
##             AFI    GroupList     FilmRank
## 1     SnowWhite    SnowWhite   Cinderella
## 2     Pinocchio   Cinderella        Bambi
## 3         Bambi Beauty&Beast    SnowWhite
## 4      LionKing     LionKing     LionKing
## 5      Fantasia        Shrek     ToyStory
## 6      ToyStory     ToyStory    Pinocchio
## 7  Beauty&Beast         Nemo        WallE
## 8         Shrek    Pinocchio           Up
## 9    Cinderella           Up         Nemo
## 10         Nemo         Nemo     Fantasia
## 11        WallE     Fantasia Beauty&Beast
## 12           Up        WallE        Shrek
```

Yes, there is a correlation. Because few movies were in similar position as AFI there exist a very weak positive correlation.

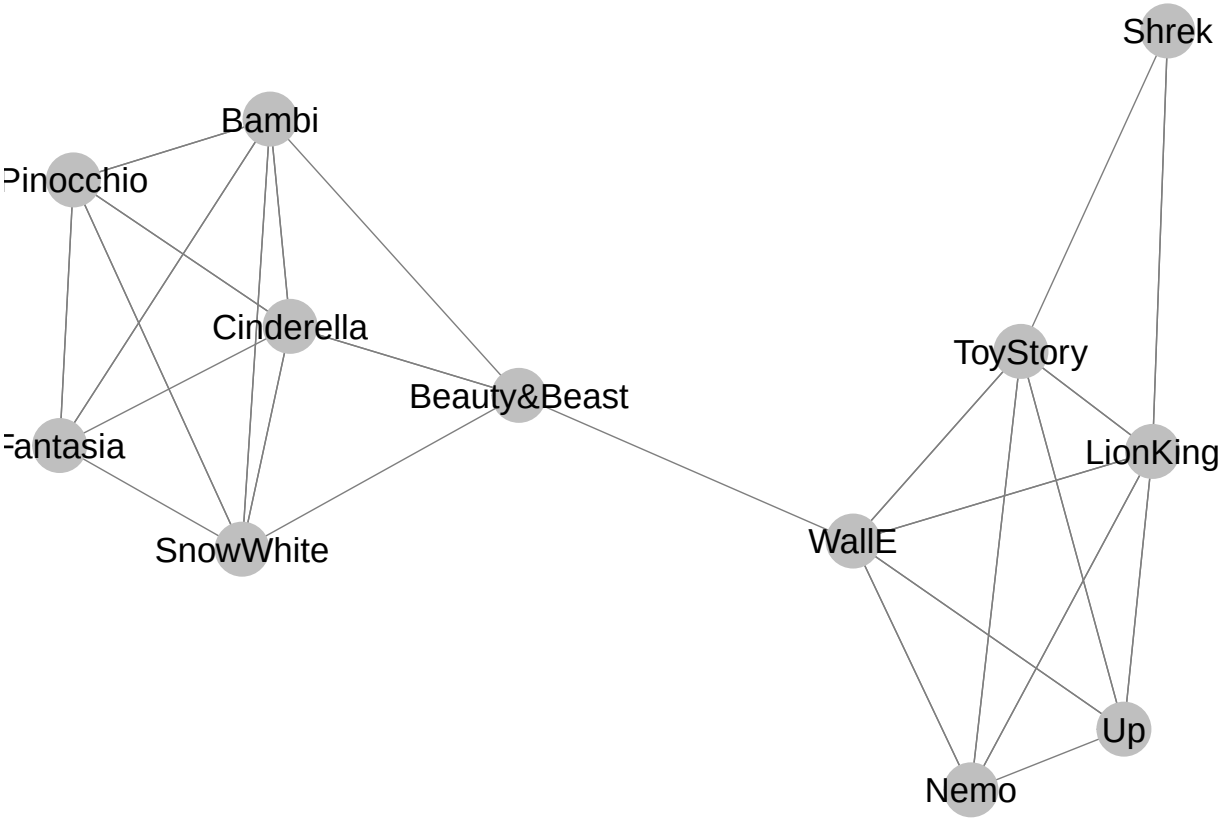
We cannot spot two movies that did not appear in the original top ten list.

8. Networks comparison

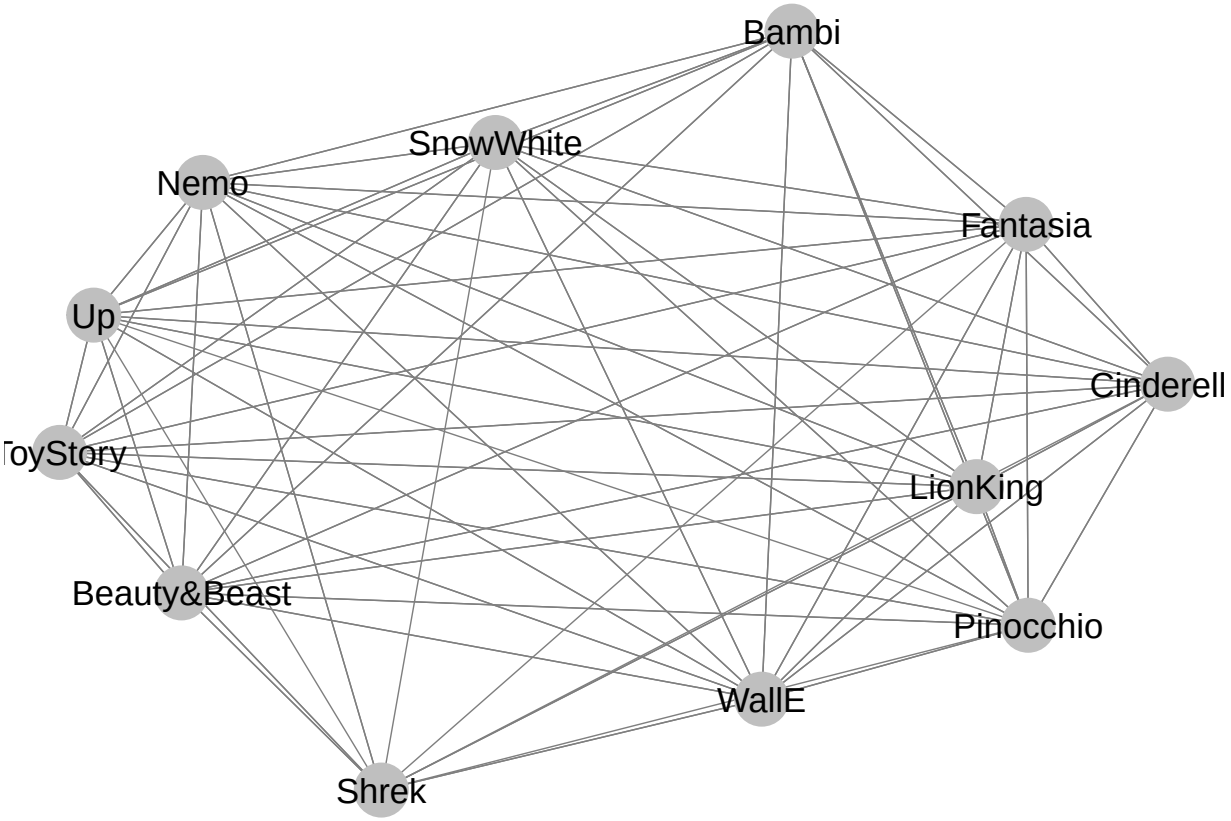
Wikipedia Network



IMDB Network



Average Network



Page-rank Network